**SQLAlchemy**

SQLAlchemy — это набор SQL-инструментов для Python и инструмент объектно-реляционного отображения (ORM), который предоставляет разработчикам всю мощь и гибкость SQL.

**ORM** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Object-Relational Mapping, [рус.](https://ru.wikipedia.org/wiki/%D0%A0%D1%83%D1%81%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *объектно-реляционное отображение*, или преобразование) — технология программирования, которая связывает [базы данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) с концепциями [объектно-ориентированных языков программирования](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), создавая «виртуальную [объектную базу](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) данных».

*# Выбор диалекта (с какой базой хотим работать)***from** sqlalchemy **import** create\_engine  
engine = create\_engine(**'sqlite:///db.sqlite3'**, echo=**True**)  
*# echo=True - будут выводиться в стандартный поток все обращения к базе данных***from** sqlalchemy **import** Column, Integer, String  
  
*# Создаем связку между базой и объектом***from** sqlalchemy.ext.declarative **import** declarative\_base  
Base = declarative\_base()  
  
**class** User(Base):  
 \_\_tablename\_\_ = **'users'** id = Column(Integer, primary\_key=**True**)  
 name = Column(String(30), nullable=**False**)  
 fullname = Column(String(30), nullable=**False**)  
  
 **def** \_\_repr\_\_(self):  
 **return "<User(name='{}', fullname='{}')>"**.format(self.name, self.fullname)  
  
*# Создать в engine таблицу  
# metadata (методанные) - описание данных таблицы, какие столбцы и  
# какие типы данных в них лежат*Base.metadata.create\_all(engine)  
  
*# Сессия через которую происходит mapping между объектом python и  
# объектом в базе данных***from** sqlalchemy.orm **import** sessionmaker  
session = sessionmaker(bind=engine)()  
  
*# Создадим первый объект типа User*user\_ivan = User(name=**'Ivan'**, fullname=**'Ivan Ivanov'**)  
  
*# Добавим объект в сессию*session.add(user\_ivan)  
  
*# Посмотрим что произошло в сессии*session.new  
  
*# Синхронизировать объект в базу данных*session.commit()  
  
*# Чтение из базы данных*query = session.query(User).filter\_by(name=**'Ivan'**)  
query.first()  
  
*# Добавить группу объектов*session.add\_all([User(name=**'Petr'**, fullname=**'Petr Petrov'**)])  
session.commit()  
  
*# Изменение объекта*user\_ivan.fullname = **'Ivan Sidorov'**session.dirty  
session.commit()  
  
*# Проброс в базу данных SQL запросов*s = session.execute(**"SELECT \* FROM users"**)  
s.first()  
  
*# Откатить сессию*session.rollback()  
  
*# Удалить объект из базы данных*session.delete(user\_ivan)  
session.commit()